



OpenSees: Analysis

Frank McKenna

2003 Users Workshop
Berkeley, CA
August 21-22, 2003

Sponsored by the National Science Foundation
through the Pacific Earthquake Engineering Research Center

Tcl Basics

- Tcl is a string based scripting language.
- Variables and variable substitution
- Expression evaluation
- Basic control structures (if , while, for, foreach)
- Procedures
- File manipulation
- Sourcing other files.

Example Tcl:

```
>set a 1
>1
>set b a
>a
>set b $a
>1
>expr 2 + 3
>5
>expr 2 + $a
>3
>set b [expr 2 + $a]
>3
```

```
>proc sum {a b} {
    return [expr $a + $b]
}
>sum 2 3
>5
>set c [sum 2 3]
>5
```

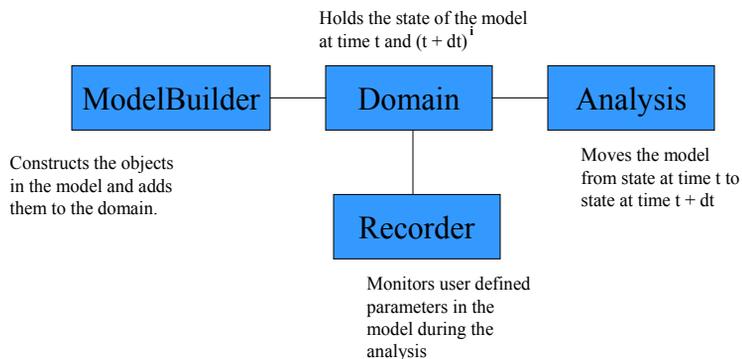
```
>set fileId [open tmp w]
>??
>puts $fileId "hello"
>close $fileId
>type tmp
hello
>
```

```
>source Example1.tcl
```

```
for {set i 1} {$i < 10} {incr i 1} {
    puts "i equals $i"
}
set sum 0
foreach value {1 2 3 4} {
    set sum [expr $sum + $value]
}
set $sum
>10
>proc guess {value} {
    global sum
    if {$value < $sum} {
        puts "too low"
    } else {
        if {$value > $sum} {
            puts "too high"
        } else { puts "you got it!" }
    }
}
> guess 9
too low
>
```



Main Abstractions in OpenSees



model command:

*Adds the modelling commands to the interpreter.

- BasicBuilder

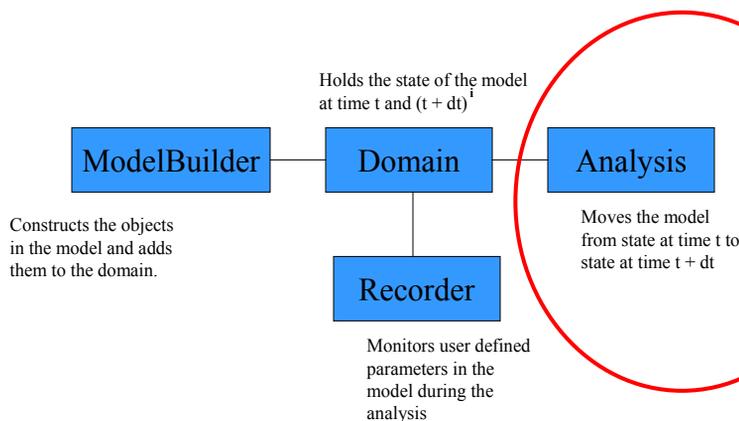
```
model BasicBuilder -ndm ndm? <-ndf ndf?>
```

This command now adds the following commands to the interpreter:

node	mass	block2D
element	fix	block3D
pattern	fixX	patch
fix	fixY	layer
equalDOF	fixZ	fiber
pattern	uniaxialMaterial	
load	nDMaterial	
eleLoad	section	
sp	geomTransf	



Main Abstractions in OpenSees



In this presentation we focus on **ANALYSIS GENERATION**

Example Analysis:

•Static Nonlinear Analysis with LoadControl

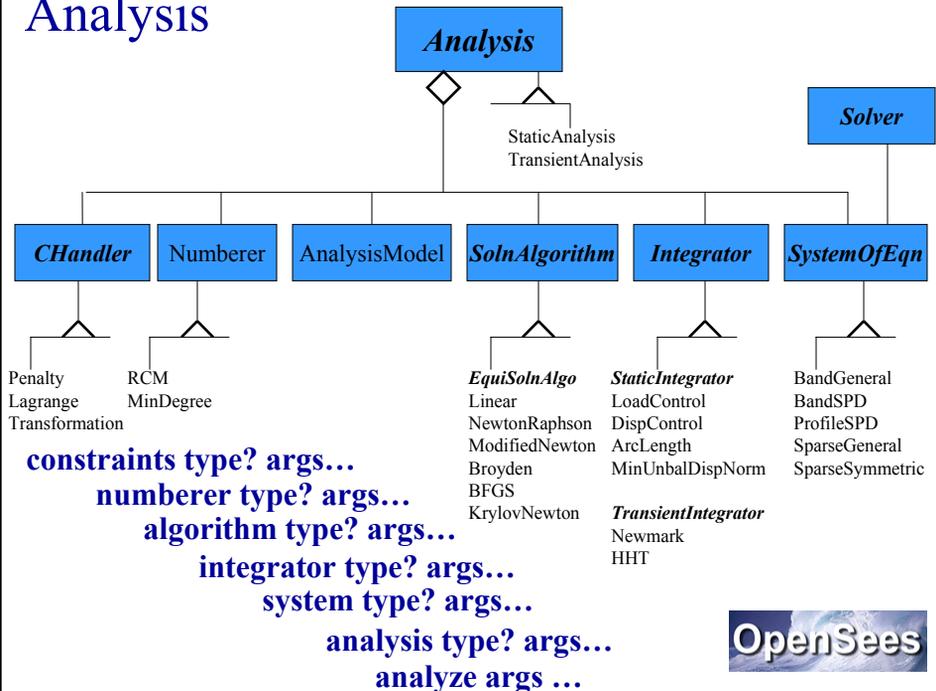
```
constraints transformation
numberer RCM
system BandGeneral
test NormDispIncr 1.0e-6 6 2
algorithm Newton
integrator LoadControl 0.1
analysis Static
analyze 10
```

•Transient Nonlinear Analysis with Newmark

```
constraints transformation
numberer RCM
system BandGeneral
test NormDispIncr 1.0e-6 6 2
algorithm Newton
integrator Newmark 0.5 0.25
analysis Transient
analyze 2000 0.01
```



Analysis



analysis command:

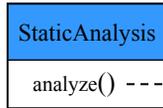
- Static Analysis

analysis static

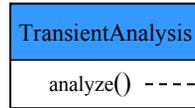
- Transient Analysis

analysis transient

- both incremental solution strategies



```
for (int i=0; i<numIncr; i++) {
  theIntegrator->newStep();
  theAlgorithm->solveCurrentStep();
  theModel->commit();
}
```



```
for (int i=0; i<numIncr; i++) {
  theIntegrator->newStep(dt);
  theAlgorithm->solveCurrentStep();
  theModel->commit();
}
```

- Eigenvalue

- general eigenvalue problem

$$(\mathbf{K}-\lambda\mathbf{M})\Phi=0$$

eigen numModes? -general

- standard eigenvalue problem

$$(\mathbf{K}-\lambda\mathbf{I})\Phi=0$$

eigen numModes? -standard

integrator command:

-determines the predictive step for time $t+\delta t$

-specifies the tangent matrix and residual vector at any iteration

-determines the corrective step based on ΔU

- Transient Integrator for Use in Transient Analysis

Nonlinear equation of the form:

$$\mathbf{R}(\mathbf{U}, \dot{\mathbf{U}}, \ddot{\mathbf{U}}) = \mathbf{P}(t) - \mathbf{F}_I(\ddot{\mathbf{U}}) - \mathbf{F}_R(\mathbf{U}, \dot{\mathbf{U}})$$

- Newmark Method

integrator Newmark $\alpha \beta$ $\langle \alpha M \beta K \beta K_{init} \beta K_{commit} \rangle$

- Hilbert-Hughes-Taylor Method

integrator Newmark α $\langle \alpha M \beta K \beta K_{init} \beta K_{commit} \rangle$

•Static Integrators for Use in Static Analysis

Nonlinear equation of the form:

$$\mathbf{R}(\mathbf{U}, \lambda) = \lambda \mathbf{P}^* - \mathbf{FR}(\mathbf{U})$$

▪Load Control

$$\lambda_n = \lambda_{n-1} + \Delta \lambda$$

integrator LoadControl Δλ

*does not require a reference load, i.e. loads in load patterns with Linear series and all other loads constant.

▪Displacement Control

$$\mathbf{U}_{j_n} = \mathbf{U}_{j_{n-1}} + \Delta \mathbf{U}_j$$

integrator DisplacementControl node dof Δλ

▪Arc Length

$$\Delta \mathbf{U}_n^T \Delta \mathbf{U}_n + \alpha^2 \Delta \lambda_n^2 = \Delta s^2$$

integrator LoadControl α Δs

▪Minimum Unbalance Displacement Norm

$$\frac{d}{d\Delta \lambda} (\Delta \mathbf{U}_n^T \Delta \mathbf{U}_n) = \mathbf{0}$$

integrator LoadControl Δλ

algorithm command:

- to specify the steps taken to solve the nonlinear equation

•Linear Algorithm

```
theIntegrator->formUnbalance();
theIntegrator->formTangent();
theSOE->solve()
theIntegrator->update(theSOE->getX());
```

algorithm Linear

•Newton-Raphson Algorithm

```
theIntegrator->formUnbalance();
do {
  theIntegrator->formTangent();
  theSOE->solve()
  theIntegrator->update(theSOE->getX());
  theIntegrator->formUnbalance();
} while (theTest->test() == fail)
```

algorithm Newton

•Modified Newton Algorithm

algorithm ModifiedNewton <-initial>

•Accelerated Modified Newton Algorithm

algorithm KrylovNewton <-initial>

constraints command:

- to specify how the constraints are enforced

$$\begin{aligned} \mathbf{U}_c &= \mathbf{C}_r \mathbf{U}_r \\ \mathbf{C} \mathbf{U} &= \mathbf{0} & [\mathbf{C}_r \ \mathbf{C}_c] \begin{bmatrix} \mathbf{U}_r \\ \mathbf{U}_c \end{bmatrix} &= \mathbf{0} \\ \mathbf{T} \mathbf{U}_r &= [\mathbf{U}_r \ \mathbf{U}_c] \end{aligned}$$

•Transformation Handler

$$\begin{aligned} \mathbf{K}^* \mathbf{U}_r &= \mathbf{R}^* & \mathbf{K}^* &= \mathbf{T}^T \mathbf{K} \mathbf{T} \\ & & \mathbf{R}^* &= \mathbf{T}^T \mathbf{R} \end{aligned} \quad \boxed{\text{constraints Transformation}}$$

in OpenSees currently don't allow retained node in one constraint to be a constrained node in another constraint

•Lagrange Handler

$$\begin{bmatrix} \mathbf{K} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{R} \\ \mathbf{Q} \end{bmatrix} \quad \boxed{\text{constraints Lagrange}}$$

•Penalty Handler

$$[\mathbf{K} + \mathbf{C}^T \boldsymbol{\alpha} \mathbf{C}] \mathbf{U} = [\mathbf{R} + \mathbf{C}^T \boldsymbol{\alpha} \mathbf{Q}] \quad \boxed{\text{constraints Penalty } \alpha_{sp} \text{? } \alpha_{mp} \text{?}}$$

system command:

- to specify how matrix equation $\mathbf{K}\mathbf{U} = \mathbf{R}$ is stored and solved

•Profile Symmetric Positive Definite (SPD)



system ProfileSPD

•Banded Symmetric Positive Definite



system BandSPD

•Sparse Symmetric Positive Definite



system SparseSPD

•Banded General



system BandGeneral

•Sparse Symmetric



system SparseGeneral

system Umfpack

numberer command:

- to specify how the degrees of freedom are numbered

- Plain Numberer

nodes are assigned dof arbitrarily

numberer Plain

- Plain Numberer

nodes are assigned dof using the
Reverse Cuthill-McKee algorithm

numberer RCM

test command:

- to specify when convergence has been achieved

all look at system: $\mathbf{KU} = \mathbf{R}$

- Norm Unbalance

$\sqrt{\mathbf{R}^T \mathbf{R}} < \mathbf{tol}$ *test NormUnbalance tol? numIter? <flag?>*

- Norm Displacement Increment

$\sqrt{\mathbf{U}^T \mathbf{U}} < \mathbf{tol}$ *test NormDispIncr tol? numIter? <flag?>*

- Norm Energy Increment

$\frac{1}{2} (\mathbf{U}^T \mathbf{R}) < \mathbf{tol}$ *test NormEnergyIncr tol? numIter? <flag?>*

analyze command:

- to perform the static/transient analysis

•Static Analysis

StaticAnalysis

analyze() ----

```
for (int i=0; i<numIncr; i++) {  
  theIntegrator->newStep();  
  theAlgorithm->solveCurrentStep();  
  theModel->commit();  
}
```

analyze numIter?

•Transient Analysis

TransientAnalysis

analyze() -----

```
for (int i=0; i<numIncr; i++) {  
  theIntegrator->newStep(dt);  
  theAlgorithm->solveCurrentStep();  
  theModel->commit();  
}
```

analyze numIter? Δt?

Example Analysis:

•Static Nonlinear Analysis with LoadControl

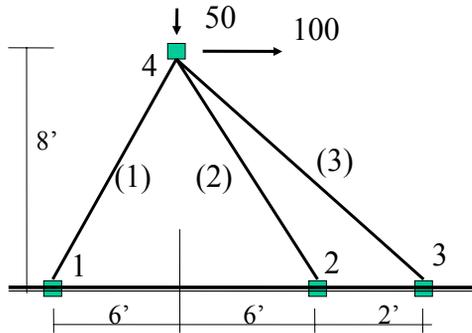
```
constraints transformation  
numberer RCM  
system BandGeneral  
test NormDispIncr 1.0e-6 6 2  
algorithm Newton  
integrator LoadControl 0.1  
analysis Static  
analyze 10
```

•Transient Nonlinear Analysis with Newmark

```
constraints transformation  
numberer RCM  
system BandGeneral  
test NormDispIncr 1.0e-6 6 2  
algorithm Newton  
integrator Newmark 0.5 0.25  
analysis Transient  
analyze 2000 0.01
```

OpenSees

Example 1: Simple Truss



	E	A
1	3000	10
2	3000	5
3	3000	5

OpenSees